

## AMENDMENT

### Amendments to the Claims:

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Currently Amended):     A ~~computer program product~~ machine-readable medium carrying one or more sequences of instructions ~~for execution by a server computer~~ for dynamically generating a wrapper object, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of comprising:

~~computer code for~~ receiving a vendor defined class object and a superclass;

~~computer code for~~ performing reflection on the vendor class to obtain vendor specific extension methods defined within the vendor class;

~~computer code for~~ generating a wrapper class as a subclass of the superclass, wherein the wrapper class comprises at least one of the vendor specific extension methods from the vendor class;

~~computer code for~~ instantiating the wrapper class, ~~the instantiating including~~ generating a wrapper object as an instance of the wrapper class by instantiating the wrapper class; and

~~computer code for~~ associating the vendor object with the wrapper object, ~~thereby enabling specific treatment of vendor objects~~ providing the wrapper object to an application program, thereby providing the application program with access to vendor specific extension methods.

2. (Currently Amended):     The ~~computer program product~~ machine-readable medium of claim 1 wherein the wrapper object is dynamically generated at runtime.

3. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 1 wherein the superclass is one of a pre-existing JDBC, JMS, or connector class.

4. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 1 wherein the superclass includes logic to handle server side tasks.

5. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 1 wherein the wrapper class is generated in bytecode.

6. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 5 wherein bytecode is generated for vendor methods not implemented in the superclass.

7. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 5 wherein the bytecode is generated using hot code generation.

8. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 1 wherein providing the wrapper object to an application program, ~~comprises providing enables~~ the application program to access to standard features defined by the superclass and non-standard vendor extensions defined by the vendor defined class.

9. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 8, wherein the standard features are J2EE features.

10. (Currently Amended): A ~~computer program product~~ machine-readable medium carrying one or more sequences of instructions ~~for execution by a server computer~~ for processing an invocation using at a dynamically generated wrapper, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of comprising:

~~computer code for receiving, from an application program, an invocation call by a wrapper object, the invocation call directed to a wrapped vendor object by an application program;~~

~~computer code for initiating pre-processing by the wrapper object;~~

~~computer code for calling the wrapped vendor object by the wrapper object;~~

~~computer code for receiving a result from the wrapped vendor object by the wrapper object;~~

~~computer code for initiating post-processing by the wrapper object; and~~

~~computer code for provide~~ providing the result to the application program, thereby enabling specific treatment of vendor objects thereby enabling the application program to access vendor specific extension methods of the wrapped vendor object.

11. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 10 wherein ~~the initiating pre-processing including~~ includes calling a pre-invocation handler.

12. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 10 11 wherein the pre-invocation handler is configured to execute server-side code.

13. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 12 wherein the server-side code includes global transaction processing code.

14. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 10 wherein initiating post-processing ~~including~~ includes calling a post-invocation handler.

15. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 14 wherein the post-invocation handler is configured to perform post-processing server side tasks.

16. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 15 wherein the post-processing server-side tasks include global transaction management.

17. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 1 wherein ~~associating the vendor object with the wrapper object~~ providing the wrapper object to an application program enables the ~~vendor object in a different manner as compared with non-vendor objects~~ application to access wrapped vendor objects without requiring a relinking of the application and a vendor software package.

18. (Currently Amended): The ~~computer program product~~ machine-readable medium of claim 10 wherein calling the wrapped vendor object ~~by the wrapper object~~ enables the wrapped vendor object to be processed ~~in a different manner as compared with non-vendor objects~~ by the application without requiring a relinking of the application and a vendor software package.

19. (Cancelled).

20. (Cancelled).

21. (New) A machine-readable medium carrying one or more sequences of instructions for enabling an application program to interface with a vendor application, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving a vendor provided class used to interface with third party software;  
preparing a wrapper object for interfacing with vendor specific extension methods of the vendor provided class by reflecting the vendor provided class and a superclass to form a wrapper class from which the wrapper object is instantiated; and  
providing the wrapper object to the application program, thereby enabling the application program capability to access vendor specific extension methods of the vendor application using the wrapper object.

22. (New) A machine-readable medium carrying one or more sequences of instructions for processing an invocation at a dynamically generated wrapper enabling an application program to interface with a vendor application, which instructions, when executed by one or more processors, cause the one or more processors to carry out the steps of:

receiving, from an application program, an invocation call directed to a wrapped vendor object;  
calling the wrapped vendor object;  
receiving a result from the wrapped vendor object; and  
providing the result to the application program, thereby enabling the application program to access vendor specific extension methods of the wrapped vendor object.